

In the Claims:

Please amend Claims 1, 4, 11, 21 and 31-39, all as shown below. Applicant respectfully reserves the right to prosecute any originally presented or canceled claims in a continuing or future application.

1. (Currently Amended) A system for loading Enterprise Java Bean software ~~applications~~ application components in an application server, the system comprising:

a computer including one or more microprocessors;

an application server for executing [[a]] an Enterprise Java Bean software application thereupon, wherein said Enterprise Java Bean software application has a plurality of modules and classes associated therewith;

an application configuration file associated with said Enterprise Java Bean software application, wherein said configuration file includes a tag layout and application ~~class-loader~~ classloader structure elements that determine [[the]] a hierarchy of modules and classes of the Enterprise Java Bean software application to be loaded into the application server, and that define a namespace separation between different modules and classes; and

a deployment logic that

parses the configuration file,

recognizes the modules and classes specified therein, and

retrieves those modules and classes from a computer readable medium in a manner consistent with the tag layout in the configuration file~~[[; and]]~~.

wherein upon receiving a request to load some or all of the modules and classes of the Enterprise Java Bean software application, the system constructs an application container at the application server with the classes and modules, in the order in which the classes and modules were retrieved, to create a ~~hierarchical class-loader~~ hierarchy of classloaders consistent with the tag layout, and to maintain the namespace separation between the different modules and classes.

2. (Original) The system of claim 1 further comprising a user interface that allows a software developer to specify a subset of said modules to be deployed.

3. (Previously Presented) The system of claim 1 wherein said modules are any of Enterprise Java Bean components, classes, or implementations.

4. (Currently Amended) The system of claim 2 wherein said user interface allows the software developer to specify a redeploy command that instructs the system that said subset should be redeployed, wherein said redeploy command specifies an Enterprise Java Bean software application name and a module associated with the application.

5. (Original) The system of claim 4 wherein said redeploy command includes a list of modules relative to the root of the application to be deployed, for redeployment of said modules.

6. (Previously Presented) The system of claim 5 wherein the module is any of Enterprise Java Bean components, class, or implementations.

7-9. (Canceled).

10. (Previously Presented) The system of claim 1 wherein the server provides multiple Enterprise Java Bean software applications, each with their own hierarchy of classloaders.

11. (Currently Amended) A method for loading Enterprise Java Bean application software components, the method comprising the steps of:

providing an application server with [[a]] an Enterprise Java Bean software application thereupon, wherein said Enterprise Java Bean software application has a plurality of modules and classes associated therewith;

associating an application configuration file with said Enterprise Java Bean software application, wherein said configuration file includes a tag layout and application ~~class-loader~~ classloader structure elements that determine [[the]] a hierarchy of modules and classes of the Enterprise Java Bean software application to be loaded into the application server, and that define a namespace separation between different modules and classes; and

parsing the configuration file, recognizing the modules and classes specified therein, and retrieving those modules and classes from a computer readable medium in a manner consistent with the tag layout in the configuration file; [[and]]

wherein upon receiving a request to load some or all of the modules and classes of the Enterprise Java Bean software application, constructing an application container at the application server with the classes and modules, in the order in which the classes and modules were retrieved, to create a ~~hierarchical class-loader~~ hierarchy of classloaders consistent with the tag layout, and to maintain the namespace separation between the different modules and

classes.

12. (Original) The method of claim 11 further comprising a user interface that allows a software developer to specify a subset of said modules to be deployed.

13. (Previously Presented) The method of claim 11 wherein said modules are of Enterprise Java Bean components, classes, or implementations.

14. (Previously Presented) The method of claim 12 wherein said user interface allows the software developer to specify a redeploy command that instructs the method that said subset should be redeployed, wherein said redeploy command specifies an Enterprise Java Bean software application name and a module associated with the application.

15. (Original) The method of claim 14 wherein said redeploy command includes a list of modules relative to the root of the application to be deployed, for redeployment of said modules.

16. (Previously Presented) The method of claim 15 wherein the module is any of Enterprise Java Bean components, class, or implementations.

17-19. (Canceled).

20. (Previously Presented) The method of claim 11 wherein the server provides multiple Enterprise Java Bean software applications, each with their own hierarchy of classloaders.

21. (Currently Amended) A computer readable medium including instructions stored thereon which when executed cause the computer to perform the steps of:

providing an application server with [[a]] an Enterprise Java Bean software application thereupon, wherein said Enterprise Java Bean software application has a plurality of modules and classes associated therewith;

associating an application configuration file with said Enterprise Java Bean software application, wherein said configuration file includes a tag layout and application ~~class-loader~~ classloader structure elements that determine [[the]] a hierarchy of modules and classes of the Enterprise Java Bean software application to be loaded into the application server, and that define a namespace separation between different modules and classes; and

parsing the configuration file, recognizing the modules and classes specified therein, and retrieving those modules and classes from a computer readable medium in a manner consistent with the tag layout in the configuration file; [[and]]

wherein upon receiving a request to load some or all of the modules and classes of the Enterprise Java Bean software application, constructing an application container at the application server with the classes and modules, in the order in which the classes and modules were retrieved, to create a hierarchical class loader hierarchy of classloaders consistent with the tag layout, and to maintain the namespace separation between the different modules and classes.

22. (Original) The computer readable medium of claim 21 further comprising a user interface that allows a software developer to specify a subset of said modules to be deployed.

23. (Previously Presented) The computer readable medium of claim 21 wherein said modules are any of Enterprise Java Bean components, classes, or implementations.

24. (Previously Presented) The computer readable medium of claim 22 wherein said user interface allows the software developer to specify a redeploy command that instructs the system that said subset should be redeployed, wherein said redeploy command specifies an Enterprise Java Bean software application name and a module associated with the application.

25. (Original) The computer readable medium of claim 24 wherein said redeploy command includes a list of modules relative to the root of the application to be deployed, for redeployment of said modules.

26. (Previously Presented) The computer readable medium of claim 25 wherein the module is any of Enterprise Java Bean components, class, or implementations.

27-29. (Canceled).

30. (Previously Presented) The computer readable medium of claim 21 wherein the server provides multiple Enterprise Java Bean software applications, each with their own hierarchy of classloaders.

31. (Currently Amended) The system of claim 1 wherein the application ~~class-loader~~ classloader structure allows for nesting of one or more application ~~class-loader~~ classloader structure elements, and wherein the outermost element of the application ~~class-loader~~ classloader structure indicates the application ~~class-loader~~ classloader.

32. (Currently Amended) The system of claim ~~[[1]]~~ 2 wherein the system allows a software developer to specify a redeploy command that instructs the system that said subset should be redeployed, wherein said redeploy command specifies an ~~enterprise~~ Enterprise Java Bean software application name and a module associated with the application, and wherein said redeploy command includes a list of modules relative to the root of the application to be deployed, for redeployment of said modules.

33. (Currently Amended) The system of claim 1 wherein the application server supports multiple Enterprise Java Bean software applications, each with their own hierarchy of application classloader~~[[-]]~~ structure elements.

34. (Currently Amended) The method of claim 11 wherein the application ~~class-loader~~ classloader structure allows for nesting of one or more application ~~class-loader~~ classloader structure elements, and wherein the outermost element of the application ~~class-loader~~ classloader structure indicates the application ~~class-loader~~ classloader.

35. (Currently Amended) The method of claim ~~[[11]]~~ 12 wherein the system allows a software developer to specify a redeploy command that instructs the system that said subset should be redeployed, wherein said redeploy command specifies an ~~enterprise~~ Enterprise Java Bean software application name and a module associated with the application, and wherein said redeploy command includes a list of modules relative to the root of the application to be deployed, for redeployment of said modules.

36. (Currently Amended) The method of claim 11 wherein the application server supports multiple Enterprise Java Bean software applications, each with their own hierarchy of application classloader~~[[-]]~~ structure elements.

37. (Currently Amended) The computer readable medium of claim 21 wherein the application ~~class-loader~~ classloader structure allows for nesting of one or more application

~~class-loader~~ classloader structure elements, and wherein the outermost element of the application ~~class-loader~~ classloader structure indicates the application ~~class-loader~~ classloader.

38. (Currently Amended) The computer readable medium of claim [[21]] 22 wherein the system allows a software developer to specify a redeploy command that instructs the system that said subset should be redeployed, wherein said redeploy command specifies an ~~enterprise~~ Enterprise Java Bean software application name and a module associated with the application, and wherein said redeploy command includes a list of modules relative to the root of the application to be deployed, for redeployment of said modules.

39. (Currently Amended) The computer readable medium of claim 21 wherein the application server supports multiple Enterprise Java Bean software applications, each with their own hierarchy of application classloader[[-]] structure elements.